Joint EU-US HEPCAL Response

V8

Editors: Fabrizio Gagliardi, Ruth Pordes

V4&8:Updated by Marcus Hardt on October 23rd, Nov 6th

V5&6&7: Updated by Ruth Pordes, October 31s/t; Nov1st,Nov 5th

1 INTRODUCTION

This document summarizes the response of the European Data Grid project and the US Physics Grid projects to the Requirements and Capabilities document presented to the SC2 by the High Energy Physics Application RTAG. Detailed reponses from EDG and the US are available at

http://www.nikhef.nl/user/templon/DataGrid-08-TEN-0202-1-8.doc and

http://www.hicb.org/glue/ushepcal_response_v3.doc respectively.

The mandate and objectives of this RTAG working group were to:

- *Identify and describe a set of high-level use cases of Grid technology common to the four experiments;*
- *Possibly identify and describe which use cases will be specific for the different experiments;*
- *Identify a set of common requirements for Grid middleware;*

*....These use cases should serve to the middleware developers (both in US and in Europe) to guide their work and to the experiments as a platform to perform Grid interoperability studies. The end product should help in the development of a common set of services for the four LHC experiments to be used on the timescale of the LHC exploitation***.**

1.1 JOINT RESPONSE

We have the following general response to the RTAG HEPCAL use cases:

1. There are no architectural constraints seen from the grid middleware perspective to meet the necessary use cases and implementations for the first LCG Production Grid deployment.

2. The very basic requirements of the HEPCAL use cases are met by the two middleware implementations considered: EDG and VDT.
3. While each of the grid middleware releases claims *functionality* is or will be implemented, we find that many more steps were required, or the user was required to specify much more information than was specified in the HEPCAL use case.
4. The boundary between what is the responsibility of the experiments and the grid middleware projects is different in the US than in the European projects. The higher level services described in many of the HEPCAL use cases are less well advanced as common services from US Grid Middleware providers. Many of these are currently seen as the responsibility of the experiment data management projects – with common approaches being sponsored and facilitated by agency (DOE, NSF) funded projects. Chimera is an example of such a higher level service. The first version is being packaged as part of VDT today and as it becomes hardened and production quality will be an integral part of the core services.
5. Underlying Grid Technologies: Both EDG and VDT rely on core grid technologies from the Globus and Condor projects. These service and software providers are providing these technologies to general application communities as well as the EDG and VDT projects. Globus and Condor provide the core of the development and support for VDT. They collaborate with and provide services to the EDG middleware releases.
6. We need to distinguish which use cases are executed directly by the physicist and which use cases can be called by their jobs, and which either actor can call.
7. It is essential that not only Use Cases but well defined Service functionality and interfaces and where possible protocols are defined to allow development and integration efforts to proceed. Without these the system will have high maintenance, integration and support costs in the delivery of integrated production services to the experiments

1.2 GENERAL REQUIREMENTS

However we feel that the charge given to the RTAG misses some general requirements for grid middleware necessary for an LCG-1 production grid:

1. Quantitative performance metrics of functionality capabilities – rates, sizes, reliability etc.
2. Specification of the underlying security model – (e.g. For example who can read a dataset? Is it just the creator? Anyone in the VO? A list of users that the creator specifies? Are there specific groups? Maybe we need a use case to change the permissions on a file?). There should be more use cases developed to address creation and management of Virtual Organizations.
3. Error reporting, handling, troubleshooting and diagnosis use cases. Additionally the level of specificity and use cases of devious flows is lacking and need better specification throughout to meet the needs of actual production deployment.
4. System and site administrator use cases, System Resource Management use cases.
5. Software deployment, versioning, configuration (the software publishing use case is very incomplete)..

1.3 RECOMMENDATIONS FOR NEXT STEPS

We make the following recommendations:

1. An RTAG on Virtual Data Requirements to include discussions with the technical teams on the experiments and the grid projects.
2. An RTAG on Grid Administration to explore the interaction between Site (and other) Administrators and the Grid. There should be a meeting between LCG Fabric and Regional center Fabric/Facility groups to discuss security, fabric-grid interface, grid administration and operational support.

3. An RTAG on catalogs and experiment meta-data databases - the scope, requirements and schedules. (this would affect both the LCG GTA and Applications areas).
4. The US Physics Grid projects should conduct a response to the use cases by people "in the trenches" deploying and using the grid applications and middleware.
5. Each middleware project should produce a test-case (making use of middleware APIs) for each use case, starting with use cases addressed in Fall 2002 and continuing with use cases addressed by software to be delivered in Spring 2003, followed by future releases.
6. Sponsorship by LCG of more formal relationship and dependency between VDT (or follow on) and EDG (or follow on).
7. Higher layer use cases should be defined and categorized e.g. production, analysis.
8. Other grid middleware projects are contributing to the LCG solutions especially in the US DOE science grid, NSF National Middleware Initiative. An understanding of the interactions with these projects, the mapping of their capabilities to the HEPCAL use cases would be useful.
9. As commercial implementations of grid middleware are available an evaluation of their functionality with respect to the HEPCALrequirements should be undertaken by the LCG
10. Revisiting use cases will be relevant when there is significant user experience with Data Challenges have been done. Reconvening the HEPCAL RTAG in the middle to late 2003 would be appropriate.
11. The LCG, EDG, US physics grid projects and RTAG groups should maintain an ongoing collaboration and common work to further elaborate the documented use cases.

2 SUMMARY OF RESPONSES TO INDIVIDUAL USE CASES

The table below gives an overall summary of ability of the 2 main HENP Grid Middleware Distributions – the European DataGrid and US Virtual Data Toolkit – to meet the requirements listed in the HEPCAL document. We note that there are other middleware distributions that could have been included but were not considered here – notably the Globus toolkit per se and NorduGrid. We believe this is appropriate as the EDG and VDT mandates specifically include delivery and support of middleware to the HENP experiments.

The table below groups the use cases into areas of capability: Authorization, Authentication and VO Infrastructure.; Meta-Data, Data Management and Access; Virtual Data Management and Access; Job Management and Scheduling; Application Services and Higher Level Tools. The comments of the EDG came from individual work packages where number of use cases treated by each WP (43 use cases in total) were: WP1 – 19 ; WP2 – 19; WP5 – 8; WP4 – 7; WP3 – 2; Architecture Consultant – 33.

We are targetting our response to the experiment requirements as understood for the LCG-1 milestone in Spring 2003.

2.1 SUMMARY OF USE CASE IMPLEMENTATION

In the tables below:

[1] – SRB, SAM, MAGDA provide some subset of these services to some application groups in the US. These products are integrated to some of the common middleware services of Globus and Condor, and work is proceeding in most of the projects to increase the integration and interoperability capabilities.

[2] – These use cases need fuller specification including information from experiments applications, data and system models.

JT - Jeff Templon

ATFAC – EDG Architecture Group

If columns are empty they are removed to help retrieve space for the comments.

Column C - Compatability between EDG and VDT

2.1.1 AAA and VO

Use Case	**EDG 1.2**	**VDT 1.1.3**	**EDG 2 planned**	**VDT Spring 2003**	**C**	**Comments**
AAA and VO						
Obtain Grid Authorization	Yes	Yes			Yes	Work with DOESG
Revoke Grid Authorization	Yes			Yes	Yes	
Grid Login	Yes	Partial	Roles, expiration issues		Yes	Both kits provide the same deficiencies: Both extensions and both additional requirements are not fullfilled. Bob Jones: it is more appropriate to say "yes" for EDG 1.2 since it does more than VDT. (I don't think VDT uses myproxy yet).
Browse Grid Resources	Basic?	Basic		Yes	Yes	Reading the two docs (EDG, VDT) VDT's "basic" (they quote the same tools as available in EDG) is also possible with EDG. EDG have just taken a more strict interpretation than the VDTguys.

2.1.2 Meta Data, Data Management and Access

Use Case	EDG 1.2	VDT 1.1.3	EDG 2 planned	VDT Spring 2003	Other US gridware	C	Comments
Meta-Data, Data Mgmt. and Access							
DS Metadata Update	Yes?	Basic			Yes - [1]	Basic	Issues about GDMP are pending. Issues of multiple implementations on US side (SAM, MAGDA, SRB) are pending. EDG will drop GDMP because it is not sufficient. The answers from both EDG and VDT seem to be way too vague for putting a yes/no answer and a 5 line comment into the document. For EDG partially due to JT's comments True for VDT: UC 5-10, 12 => Relying on table from VDT answer. EDG: UC 5, 6, 7 => Relying on marginal description from EDG answer
DS Metadata Access	Yes?	Basic		Yes	Yes – [1]	Basic	EDG: User must do all the work! EDG: Primitive (J.T.)
Dataset (DS) Registration	Yes?	Basic		Yes?	Yes – [1]	Basic	
Dataset Upload	Yes	Basic		Yes	Yes – [1]	Basic	EDG: Supported. Caveat: Compound dataset are left to the user
User defined Catalogue Creation		External		Yes?	Yes – [1]	Basic	VDT: Currently Experiment-specific. Technically possible VDT does not mention a planned support in their text, but the table says they will supported in Spring 2003. Perhaps this is because external software (which remains to be specified) exists that can (easily?) be integrated. EDG Has not considered yet. Comparing the technical infrastructure EDG has only Globus and MySQL in common, Oracle and SRB might not be provided by ALL of the regional centers.
Dataset Access	Complex		Yes		Yes – [1]		Both: POSIX currently unsupported VDT: No answer, except for the hint that there's other US Gridware EDG: Implemented, but not user friendly. (WP5 will support SRM type functionalty)
DS xfer to non-Grid storage	Complex						VDT: "US Grid MW ang regional centers are defining animplementing standard interfaces". "Investigating Posix inferfaces to the disk storage layers". EDG: "Fully support use case" -- for an insider, but too complex for the user.
DS Replica Upload	Basic						EDG: No consistency checks.
DS Access Cost Evaluation			Yes				VDT: Will be prototyped over next 12 months, then incorporated => Later planned. EDG: WP2: WIll be in EDG 2. WP5: SEGetCost() implemented
Dataset Replication	Complex		Yes	??	Yes – [1]	Yes	EDG: "Implemented in 1.2" -- JT: but too many parameters again.
Physical DS Instance Deletion	Complex				Yes – [1]		VDT: The underlying storage does, there are interfaces, but we don't hear about the use case EDG: WP2: "Implemented in 1.2" -- JT: In theory yes, not in practice
Dataset Deletion					Yes – [1]		EDG: WP2: "Not planned for any release".

Use Case	EDG 1.2	VDT 1.1.3	EDG 2 planned	VDT Spring 2003	Other US gridware	C	Comments
Catalogue Deletion					Yes – [1]		VDT: "Responsibility of the Experiments" <=> "Needs to be compatible with underlying SM sys." EDG: "Not considered yet"
Read from Remote Dataset	Complex				Basic– [1]		EDG: WP2 "Implemented in rel 1.2" -> JT: Too complex WP5 "Partial transfers will be possible using the POSIX interface"
Dataset Verification			Yes		Basic– [1]		VDT: Higher level tool EDG: WP[12] Planned for rel 2.
Dataset Browsing			Yes		Yes - [1]		EDG: WP2: "Planned for rel. 2", "A web services interface to th e metadata catalogue is planned for the next release"
Browse Expt Database				Yes	Yes - [1]		VDT: [2]: "Use cases need fuller specification" EDT: WP2: "Planned for rel 2", WP2, JT: "experiment specific layer may be needed"/ ATFAC: "This is quite a big use case to even think about leaving out" . JT: "a lot more work will be needed to specify database replication". Use case seems to be too generic

2.1.3 Virtual Data Management

Use Case	EDG 1.2	VDT 1.1.3	EDG 2 planned	VDT Spring 2003	After EDG 2	VDT later plans	Comments
Virtual Data Management							
Virtual DS Declaration			Unsure due to contradictig information from WP1 and WP2	Yes		Yes	EDG and VDT, wait for results from the GriPhyn project (Chimera). Integration into EDG, VDT and experiments is not yet understood.
Virtual DS Materialization				Yes		Yes	

Both projects don't seem to have tackled this issue. More or less both projects wait for output from GriPhyn/Chimera (VDT more, EDG less)

o EDG states that the functionality is given in general but support is "Not planned for any release".

o VDT thinks, that "Virtual Data is not a requirment for LCG Production Deployment in 2003/2004".

2.1.4 Job Management and Scheduling

Use Case	EDG 1.2	VDT 1.1.3	EDG 2 planned	VDT Spring 2003	Other US gridware	C	Comments
Job Mgmt. and Scheduling							
Job Catalogue Update				Yes		Yes	EDG: WP1: "Planned, but after release 2", There's a doc in EDMS in "WP1 tasks" describing a suggestion on this UC
Job Catalogue Query		Yes	Yes	Yes		Yes	VDT: Yes, but no details EDG: WP1 Planned after EDG-2

Use Case	EDG 1.2	VDT 1.1.3	EDG 2 planned	VDT Spring 2003	Other US gridware	C	Comments
Job Submission	Yes	Yes				Yes	
Job Output Access	Complex	Basic		Yes	Baisc - [1]	Yes	VDT: EDG2 compatible" EDG: WP1: "Supported in rel 1". JT: "Can't list files, too complex, can block RB. JT to ATFAC response: RB can be filled up and thus be paralyzed. Quotas are not thought of yet, but should be considered.
Error Recovery for failed Jobs							EDG: WP1: Not planned within EDG
Job Control	dg-job-cancel	Basic		Yes			VDT: EDG2 compatible" EDG: WP1: "Not planned for rel 2, will be taken into account [..] for year 3", together with checkpointing. EDG: WP4: Extensive answer to extensive part of UC. See EDG-answer p29 EDG: ATFAC: Priority is amiguous => WP8 is waiting for proposal from WP4
Steer Job Submission	yes			Basic		Yes	EDG: Everything but cost plugin supported. That is considered for year 3
Job Resource Estimation	yes						EDG: WP1: "Implemented in rel 1"
Job Environment Modify	yes	Basic					
Job Splitting		Basic		Yes		Yes	EDG: WP1: "planned after rel 2". Improve the usecase!
Job Monitoring	Yes	Basic		Yes		Yes	VDT: "Monitoring of jobs is provided as part of the Grid Scheduling and Fabric Batch systems" "experince [..] (shows) sufficient middleware [..] for job monitoring for LCG (exists)"; EDG2 compatible" EDG: WP1: "[..] implemented in rel 1" Missing details will be added by Year 3 refinements (WP4 dependency) EDG: WP4: "Being addressed" Going through all points, discussing solution, if applicable.

2.1.5 Application Services and Higher Level Tools

Use Case	EDG 1.2	VDT 1.1.3	EDG 2 planned	VDT Spring 2003	After EDG 2	VDT later plans	Comments
Application Services and Higher Level Tools							
Production Job					Yes		Use case has to be more detailed VDT: Until UC better defined, it is out of the scope of US MW providers EDG: WP1: "planned after rel 2. ATFAC: "This use case is not ready for review"
Analysis 1			Yes				VDT: These "specifics are outside the scope of the grid middleware providers" EDG: WP1: "Planned for rel 2" . ATFAC: Needs to be distinguished from UC#dstran (36)
Data			Yes				VDT: These "specifics are outside the scope of the grid middleware

Use Case	EDG 1.2	VDT 1.1.3	EDG 2 planned	VDT Spring 2003	After EDG 2	VDT later plans	Comments
Transformation							providers" EDG: WP1: "Planned for rel 2. JT: Unclear if WP1's one-job-for-each-subtask-solution matches HEP's view of the UC. AFTAC: One referred UC is missing. Suggestions/Discussion on how to subdivide the UC.
Conditions Publishing	No middleware components						VDT: "Outside the current scope of MW projects" table: "Use cases need fuller specification"
Software Publishing		Basic		Yes		Yes	VDT: Comes with PACMAN (from US ATLAS) EDG: WP4: Although the automated per-fabric installation of system middleware and application software package is addressed, the installation of software by users is not supported. That is because of Administrative privileges required. JT: Another approach is required since different apps can conflict.
VO-wide Resource Reservation			API	Basic	Yes	Yes	VDT: “VO procedures [..] (are) currently lacking in funded effort" EDG: WP4: "Being addressed" WP1: "API for rel 2, integration into RB after rel 2”. Restrictions of type of resources depent on WP[457]. ATFAC: "Need to resolve open question"
VO Resource Allocation to Users				Basic	Need more information	Yes	VDT: "VO procedures [..] (are) currently lacking in funded effort" EDG: ATFAC: "Much more detail required"
Simulation Job			Yes				VDT: "Provided by the integration of Experiment specific scripts/programs and job and data management" EDG: WP1: "Planned for rel2. see dstran use case
Exp’t Software Dev for Grid					Yes	Yes	EDG: WP1: "Planned for release 2". ATFAC: Come back to this UC after software publishing questions are answered

2.2 REQUIREMENTS FOR LCG-1

With the understanding that many of the use cases available are available in basic form but not necessarily in transparent or easily usable form we summarize the stated needs of the experiments for LCG-1 – as documented inLHC Computing Grid Project (LCG) Phase 1, Status of High-Level Planning, version 1 - 21 June 02.

NA - would not be available for LCG-1 in mid-2003

A - available for LCG-1 in 2002 (2) or 2003 (3).

B- available for LCG-1 in basic form in 2002 (2) or 2003 (3)

C - available for LCG-1 but in a (too) complex a form.

I – Interoperability between EDG and VDT

E – Experiment specific implementation

Use Case	
AAA and VO	
Obtain Grid Authorization	I A 2
Revoke Grid Authorization	I A 3
Grid Login	I B 2
Browse Grid Resources	I B 2
Meta-Data, Data Management and Access	
DS Metadata Update	B E 2/3
DS Metadata Access	B E 2/3
Dataset (DS) Registration	B E 2/3
Dataset Upload	B E 2/3
Catalogue Creation	B E 2/3
Dataset Access	C 2
DS xfer to non-Grid storage	C 2
DS Replica Upload	C 3
DS Access Cost Evaluation	NN
Dataset Replication	I C 2
Physical Instance Deletion	I B C 2
Dataset Deletion	C 2
Catalogue Deletion	C 2
Read from Remote Dataset	NA
Dataset Verification	NA
Dataset Browsing	I B 2
Browse Expt Database	E
Virtual Data Management	
Virtual DS Declaration	NA
Virtual DS Materialization	NA
Job Management and Scheduling	**Below depends somewhat on which fabric job management system is used. PBS, LSF, FBS, and Condor are supported by some but not all sites.**
Job Catalogue Update	E 3
Job Catalogue Query	I A 2
Job Submission	I A 2
Job Output Access	I C 2
Job Control	I A 2
Steer Job Submission	NA
Job Resource Estimation	NA
Job Environment Modify	NA
Job Splitting	NA
Job Monitoring	I A 2
Application Services and Higher Level Tools	

Use Case	
Production Job	E
Analysis 1	E
Data Transformation	NA
Conditions Publishing	NA- E
Software Publishing	B 3
VO-wide Resource Reservation	B 3
VO Resource Allocation to Users	NN
Simulation Job	E
Exp't Software Dev for Grid	E

3 APPENDIX A: EXAMPLE APPLICATION-MIDDLEWARE INTERFACE SCRIPT

There are three portions of the example included below.

a) One is a basic analysis of the use case, showing how to turn the use case into the simplest possible use case for the middleware to handle.

b) The next shell script shows how to implement the use case, with as little work as possible, given the current state of the middleware. If the shell script contains many steps, this is indicative that the middleware doesn't implement the use case, it implements more "elemental" use cases that a user can glue together to achieve the desired functionality. The key here is that the user is responsible for

maintaining the glue
keeping track of any interface changes for the various MW-supplied pieces
keeping track of any extra information needed (location of info system for example)
supplying extra parameters (parameters not specified as needed in HEPCAL)

For the example presented in this mail, if the use case was really implemented, we would be able to upload a dataset to the grid via one command, and the user would not need to worry about anything that happened internal to that command.

c) A bare-bones version of an analysis that might be provided by GAG when asked "please evaluate this implementation".

3.1 ANALYSIS OF THE USE CASE

I try here to make UC#dsupload as basic as possible, meaning I as the user execute the use case in a way that places as few demands on the grid middleware as possible.

Examining UC#dsupload from HEPCAL, one can specify the minimal basic use case as follows:

1. User specifies:

a. the source file belonging to the data set [option for multiple files omitted for this basic version of the use case]
b. Information to register a dataset [see uc#dsreg for which info]

i. the LDN
ii. [optional access protocol, omitted here for basic case]
iii. [optional metadata, omitted here for basic case]

2. Extension point: the user specifies an SE where the physical file should be placed [we choose this option here since it makes the implementation more basic ... the implementation would be more complex if no SE were specified]

3, 4, 5 are executed by the Grid

6. The system confirms success and reports the LDN under which the file(s) are registered.

From a user point of view, this basic use case is:

```
  LDNretval = dsupload(source_file, LDN, targetSE)
```

A (Bourne) shell version might look like

```
  LDNretval = $(edg-dsupload -s source_file -l LDN -d targetSE)
```

3.2 INTERFACE CODE

```
#!/bin/bash

# implementation of uc#dsupload for edg 1.2 (b4 edg-replica-manager)

# assume called as

#          uc.dsupload -s source_file -l LDN -d targetSE -v VO

usage="Usage: $0 -s source_file -l LDN -d targetSE -v VO"

### Note: VO is an extra parameter above HEPCAL

### the following section is information that in this implementation
### needs to be adapted by user, whereas in HEPCAL the grid takes care
### of it.

IS_HOST=lxshare0382.cern.ch
IS_PORT=2170

while getopts ":s:l:d:v:" opt; do
  case $opt in
    s ) SOURCEFILE=$OPTARG ;;
    l ) LDN=$OPTARG ;;
    d ) TARGSE=$OPTARG ;;
    v ) VONAME=$OPTARG ;;
    \?) echo $usage
        exit 1 ;;
  esac
done

if [ \( -z "$SOURCEFILE" \) -o \( -z "$LDN" \) -o \
    \( -z "$TARGSE" \)     -o \( -z "$VONAME" \) ] ; then
   echo $usage
   exit 1
fi

# set up GDMP config info

GDMP_CONFIG_FILE=/opt/edg/etc/$VONAME/gdmp.conf ;  export GDMP_CONFIG_FILE

# determine where my VO needs to write on target SE

destpath=$(ldapsearch -h $IS_HOST -p $IS_PORT -x -b "seId=$TARGSE,o=grid" | \
          gawk -F : '/^SEvo.*'$VONAME'/ { print $3 }')

# copy file to remote location

if [ $(dirname $SOURCEFILE) = "." ] ; then
   SOURCEFILE=$(pwd)/$SOURCEFILE
fi

globus-url-copy file://$SOURCEFILE gsiftp://$TARGSE/$destpath/$LDN

gdmp_register_local_file -S $TARGSE -R -p $destpath/$LDN

sleep 10

gdmp_publish_catalogue -S $TARGSE -C
```

3.3 ANALYSIS OF THE INTERFACE CODE

The EDG 1.2 implementation of Use Case #dsupload is reviewed. The basic functionality represented by the use case, namely "transfer a dataset from non-grid storage onto the grid, and registering it with the data management service" is adequately implemented.

The implementation is quite a bit more complex than specified in the use case. In the HEPCAL use case, the user has to specify only a few things:

1. the source file (or files) which will comprise the dataset
2. the logical dataset name (LDN) to be assigned to the dataset
3. optionally an access protocol
4. optionally metadata to be associated with the file
5. optionally an SE where the dataset should initially be placed

The use case implementation supplied assumed that the user specified information to make things "as easy as possible" for the middleware. The implementation required the following command parameters:

source file name
logical dataset name
target SE name (more work is required to discover all SEs available
              and pick a good one, so we make it easy by asking the
               user).
Name of VO to which user belongs.

This additional parameter was required since there is no standard way to find out which VO a user belongs to at the moment. The WP2 tools do not include any "smart" mechanism to discover the VO.

A shell script was provided that glued some WP2 tools together into something that looked like the HEPCAL use case command. Grid users would have to include something like this shell script into their job scripts or programs, if they intend to use this use case.

In addition to the command parameters, the person using the shell script must "hard-wire" the hostname and port of the Grid information system. The WP2 tools are not capable of discovering all necessary information about remote SEs so the user must assist by querying the information system. In the HEPCAL call, the Grid does all the work of keeping track of the info system location.

Ignoring some basic steps associated with argument-parsing and trivial error trapping, the steps in the shell script are as follows:

1. query the info system to find info about the target SE
2. extract from this information the "gdmp area" associated with the user's VO -- this is where the current user is allowed to write data
3. copy the file to the remote SE / path using gridFTP (globus-url-copy)
4. contact the gdmp server at the remote SE and register the file there
5. have the remote gdmp server publish its local catalog to the Replica Catalog

Since this implementation is a shell script, users incorporating this code into their job scripts will need to carefully track changes in syntax or operation of the various commands used, and adapt their scripts accordingly. We look forward to having a "dsupload" command where all the work is managed internal to the command, and the user is relieved from worrying about "internal" details such as interface changes and service location changes.

Summary
========

The implementation based on GDMP consists of five steps rather than one.

The user must maintain two pieces of information (info system hostname and port) in order for the implementation to work. This is not specified in the use case.

The user must provide an extra piece of information (user's VO) compared to the use case.

The functionality is implemented, but the implementation is substantially more complex than specified in the use case.

This is a "best case" analysis. If one for example had omitted the targetSE parameter, several steps would need to be added.

1. query to the information system to find all SEs known to the system.

2. filter this list of SEs to those accepting the users VO

3. further filter this list to those SEs which have enough disk space to hold the source file

4. select at random one of the members of this list

At this point, one can then make the info system query to find the VO path on the SE, etc.